



## **Auto-incrémentation d'une base dysfonctionnelle de cas pour un système d'aide au diagnostic et à la réparation.**

Brigitte Chebel-Morello, Mohamed-Karim Haouchine, Nouredine Zerhouni

### **► To cite this version:**

Brigitte Chebel-Morello, Mohamed-Karim Haouchine, Nouredine Zerhouni. Auto-incrémentation d'une base dysfonctionnelle de cas pour un système d'aide au diagnostic et à la réparation.. 3ème Edition du Colloque International Francophone sur la Performance et les Nouvelles Technologies en Maintenance, PENTOM'2007., Jul 2007, Mons, Belgique. sur CD ROM - 22 p. hal-00163976

**HAL Id: hal-00163976**

**<https://hal.science/hal-00163976>**

Submitted on 19 Jul 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Auto-incrémentation d'une base dysfonctionnelle de cas pour un système d'aide au diagnostic et à la réparation

Brigitte Chebel-Morello, Mohamed Karim Haouchine, et Nouredine Zerhouni

Laboratoire d'Automatique de Besançon,  
UMR CNRS 6596.  
24 Rue Alain Savary  
25000 Besançon France.  
{karim.haouchine, bmorello, zerhouni}@ens2m.fr

---

**RÉSUMÉ.** Le raisonnement à partir de cas est une méthode d'intelligence artificielle largement utilisée dans la résolution de problème de diagnostic technique. Après avoir mis en place un système de diagnostic et de réparation dédié à un système de transfert de palette nous nous sommes intéressés à la maintenance de ce système et tout particulièrement à l'optimisation de la base de cas qui est au cœur du système et à sa remise à jour. Nous proposons dans cet article dans un premier temps d'optimiser la base de cas d'un système de raisonnement à partir de cas dédié au diagnostic de pannes et dans un deuxième temps d'enrichir la connaissance de ce système en rajoutant des cas de diagnostic non recensés d'une manière dynamique, sans altérer la structure de la base de cas mise en place. Ces 2 propositions ont été mises en place sur une plateforme de e-maintenance.

**ABSTRACT.** Case based reasoning is a methodology of intelligence artificial. Case-based Reasoning is an artificial intelligence method, widely used while solving technical diagnostic problems. After having set up a system of diagnostic and repair dedicated to a pallet transfer system, we became interested in the maintenance of this system. Especially concerning optimizing and up-dating the case base which is situated at the heart of the system. In this article, we propose for the first time to optimize the case base of a case base reasoning system seen from a dedicated diagnostic default. In the second time, to enrich the knowledge of this system by adding non- listed diagnostics case in a dynamic way, without destroying the case base structure. These 2 proposals were installation on an e-maintenance platform,

**MOTS-CLÉS:** aide au diagnostic et à la réparation, raisonnement à partir de cas, apprentissage incrémental, maintenance.

**KEYWORDS:** Diagnostic and reparation help, case-based reasoning, incremental learning, maintenance.

---

## 1. Introduction

Le **Raisonnement à Partir de Cas (RàPC)** est une approche de résolution de problème et d'apprentissage qui consiste à ré-utiliser des expériences passées stockées sous forme de cas dans une mémoire appelée base de cas (**BC**) afin de résoudre un nouveau problème (Aamodt et al., 1994).

Actuellement, plusieurs systèmes de RàPC existent et tournent avec succès INRECA (Bartsch-Spörl, 1995) et particulièrement des applications industrielles construites à partir de systèmes de développement (KAIDARA). L'utilisation du raisonnement à partir de cas (RàPC) en diagnostic est particulièrement répandue ; On compte plusieurs systèmes de diagnostic industriel traités par cette approche, à savoir :

- CaseLine et CASSIOPEE qui s'occupent de dépannage de moteurs d'avions : CaseLine (Watson et al., 1994) est un démonstrateur utilisé par British Airways pour le diagnostic de pannes et la réparation de Boeing 747-400, CASSIOPEE (Bergmann et al., 2003) qui s'intéresse aux Boeing 737 à CFM International.
- CHEKMATE dédié au diagnostic de pannes relatifs aux imprimantes industrielles par Domino UK Ltd. présenté dans (Grant et al., 1996).

Dans cette mouvante nous avons conçu, un système d'aide au diagnostic de pannes et à la réparation dédié à un système de transfert de palettes SORMEL à l'aide du raisonnement à partir de cas (Rasovska et al., 2005).

Tout système informatique arrivé à maturité nécessite en phase de fonctionnement d'être maintenu, ne serait ce que mis à jour. En effet les données informations connaissances évoluent dans le temps et doivent être réactualisées dans le système pour éviter toute dégradation dans le temps de sa performance. Dès leurs conceptions les systèmes de raisonnement à partir de cas tiennent compte de l'évolution de leur base de cas, toutefois cela ne les affranchit pas d'une étape de maintenance.

Le cycle du RàPC que l'on vous décrira rapidement au paragraphe 2 comprend cinq phases, que l'on a développées pour le diagnostic de pannes et de réparation d'un système de transfert de palette (Rasovska et al., 2005). Dans cette communication, notre contribution se situe dans les deux dernières phases de ce cycle qui sont déterminantes dans le maintien des bonnes performances du système de RàPC et dans sa capacité à intégrer de nouveaux cas. Toutefois cette intégration doit être faite sous certaines conditions afin de garantir la qualité du système.

Plusieurs auteurs spécifient que cette maintenance peut se réduire à une maintenance de la BC du système. Iglezakis soutient dans (Iglezakis et al., 2000) qu'il ne peut y avoir de maintenance de système de RàPC sans le balayage de la BC.

Que la BC est associée à la connaissance du système, que les opérations de maintenance ne sont déclenchées que grâce à elle. La BC a un rôle central ce qui explique que la majorité des travaux faits dans ce domaine, est essentiellement fondé sur la Maintenance de la Base de Cas (**MBC**). De plus les connaissances d'un système de RàPC sont liées aux cas puisque ceux-ci sont affectés par tout changement dans le système, sa consultation est la plus appropriée pour déclencher les opérations de maintenance (Zehraoui, 2004). Nous avons donc étudié la maintenance de la base de cas dans (Zehraoui, 2004). Prenant appui sur un modèle de performances de Smith qu'on a amélioré grâce à une mesure de compétence, nous avons réorganisé la base de cas et l'avons réduit aux cas les plus pertinents. Nous vous décrirons notre démarche au paragraphe 3 et nous proposons au paragraphe suivant un algorithme incrémental permettant d'intégrer un nouveau cas adhérent à cette organisation

Le paragraphe 5 est consacré à l'application à un système de transfert de palettes, des différents algorithmes présentés dans le papier

## 2. Raisonnement à Partir de Cas et diagnostic

### 2.1 Introduction

Le RàPC a pour objectif de résoudre un nouveau problème, à l'aide d'une base de cas. Un cas est une connaissance qui représente une expérience et est la description d'un épisode de résolution de problème. Il peut donc prendre des formes très diverses selon la nature de la tâche : diagnostic, planification, aide à la décision, conception, etc. Il contient au moins deux parties, une description de situation représentant un « problème » et une « solution » utilisée pour remédier à cette situation (Mille, 1995). Nous empruntons à Jean Lieber (Lieber 2007) les notations suivantes :

On note **Problèmes** l'espace des problèmes et **Solutions** l'espace de solutions

Soit  $pb \in \mathbf{Problèmes}$  et  $sol(pb) \in \mathbf{Solutions}$ . Résoudre un problème  $pb$  c'est trouver ou construire  $sol(pb)$ . Un cas est donc composé de  $(pb, Sol(pb))$ .

Un cas est stocké dans la base de cas est nommé cas source et est noté  $(srce, Sol(srce))$ .

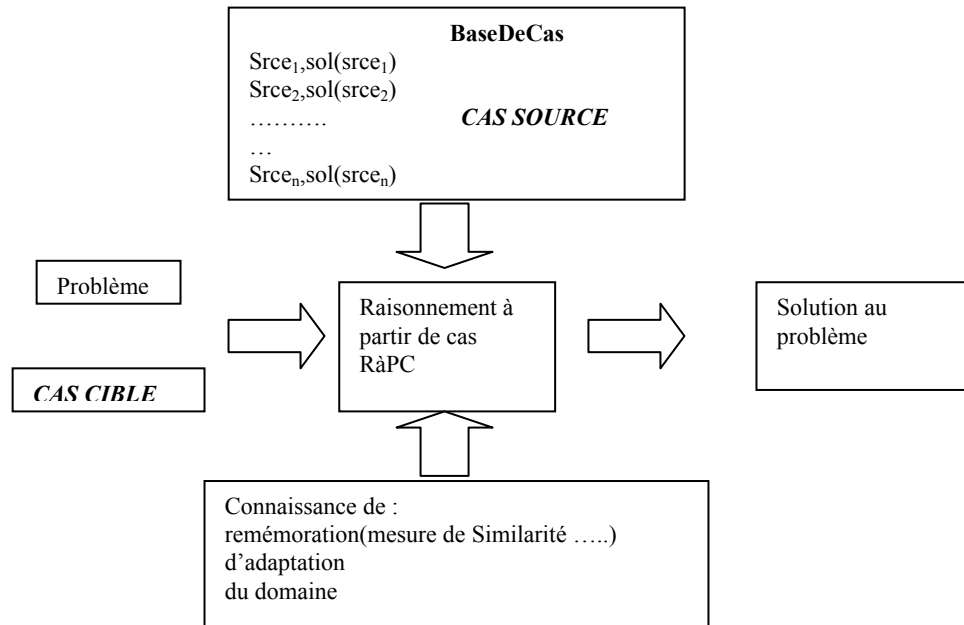
Une base de cas  $BaseDeCas$  est un ensemble fini de cas.

dont on va s'inspirer pour résoudre un nouveau cas qu'on appellera *cas cible*.

le problème à résoudre le cas cible  $\in \mathbf{Problèmes}$

Raisonnement à partir de cas fait appel à une base de cas :

RàPC :  $(cible, BaseDeCas) \longrightarrow Sol(cible) \in \mathbf{Solutions}$



**Figure 1.** *Le raisonnement à partir de cas*

En effet, le RàPC permet à partir d'une description de panne d'en retrouver la ou les causes et de proposer une action pour une éventuelle intervention de maintenance. La structure des cas va dépendre des domaines d'utilisation et des tâches à accomplir. Adaptée au diagnostic, la structure du cas est donc la suivante :

Problème  $\longleftrightarrow$  Symptômes (description d'une situation de diagnostic)  
 Solution  $\longleftrightarrow$  Origines (plusieurs possibles) + Actions (stratégie de maintenance).

Un cas décrit donc une situation particulière et contient plusieurs attributs définis par leurs valeurs, propres à la situation. Un exemple de la structure d'un cas dédié au transfert SORMEL est montré sur la figure 2.

|   | Descripteurs        | Attributs-valeurs  |
|---|---------------------|--|
| C | Problème (symptôme) |  |
| A |                     | <ul style="list-style-type: none"> <li>- zone : anneau principal</li> <li>- sous-zone : entrée</li> </ul>  |
| S |                     | <ul style="list-style-type: none"> <li>- composant-équipement : pousseur</li> <li>- présence palette : oui</li> </ul>  |
| l |                     | <ul style="list-style-type: none"> <li>- type et détecteur principal : D3 = 1</li> <li>- stoppeur et son état : S2 = 0</li> <li>- type et état du détecteur du contexte : pousseur ne revient pas</li> </ul> |
|   | Solution            | <ul style="list-style-type: none"> <li>- Classe : pousseur</li> <li>- Diagnostic : pousseur bloqué</li> <li>- Action : débloquer pousseur</li> </ul>   |

Figure 2. Exemple d'une structure d'un cas

## 2.2 Cycle du RàPC

Le RàPC est doté d'un cycle comportant généralement cinq phases comme le montre la figure 3 (Aamodt et al, 1994) :

- *L'Elaboration* du cas cible.
- *La Remémoration* des cas sources similaires au cas cible.
- *L'adaptation* de la solution au problème du cas cible inspirée de la solution du cas source le plus similaire.
- *La Révision* de la solution proposée.
- *La Mémorisation (apprentissage)* du nouveau cas résolu si ce dernier n'est pas recensé dans la base de cas, c'est l'occasion d'enrichir la base de cas.

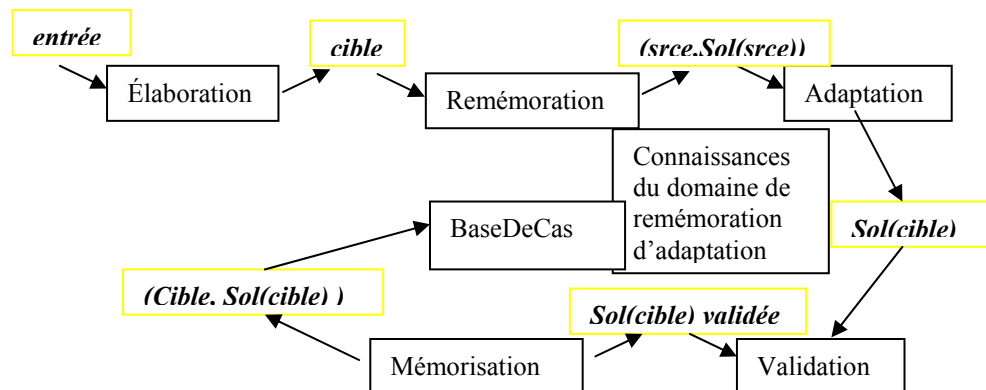


Figure 3. Le cycle de raisonnement à partir de cas (Aamodt et al., 1994)

### 2.2.1 L'élaboration du cas

L'élaboration d'un nouveau problème ou sa description se fait à chaque utilisation du système ; à partir d'une requête, la partie description du cas est expliciter dans un premier temps avec les informations de la requête et est complété par des techniques d'induction, de calcul ou par des collectes de données permettant d'élaborer un cas ayant une structure similaire au cas de la base de cas.

### 2.2.2. La Remémoration du cas

La remémoration des cas les plus similaires au cas à résoudre permet d'obtenir comme l'indique la figure 2 un ou plusieurs cas (*srce*,*Sol(srce)*) de la *BaseDeCas*

Cette phase est basée sur une mesure de similarité entre les cas de la base de cas appelés sources et le cas à résoudre appelé cible.

$S$  : Problèmes x Problèmes (0 ;1) Soit  $S_{min}$  . un seuil prédéfini

2 problèmes sources et cibles sont similaires si  $S(srce,cible) \geq S_{min}$  .

Ces mesures de similarité cherchent des correspondances entre descripteurs de cas source et cible et peuvent souvent tenir compte de la pondération éventuelle des descripteurs dans un cas. Les cas similaires sont ensuite retrouvés à l'aide d'un algorithme de recherche : les K plus proches voisins (KPPV) (Bagui et al., 1995), les approches inductives, les réseaux de recherche de cas (Case Retrieval Nets – CRN) (Lenz, 1998).

On peut également utiliser des mesures de dis similarité

### 2.2.3. L'adaptation du cas

L'adaptation sert à faire évoluer la solution du cas source similaire (*srce*,*Sol(srce)*) en solution plus adapté au problème cible *Sol(cible)*.

D'après (Lee, 2003), pour améliorer le fonctionnement d'un système de RàPC dans la phase d'adaptation nous avons deux possibilités. Premièrement, éviter d'adapter un cas en améliorant la remémoration et donc directement la base de cas pour que la solution du cas similaire soit acceptable et utilisable pour un nouveau cas. La deuxième possibilité est de chercher une meilleure technique d'adaptation. Les principaux types d'adaptation automatique sont les suivantes (cf. Wilke, 1998):

- *l'adaptation générative*, où on dispose de toutes les connaissances pour résoudre le problème à partir de zéro. Le cas retrouvé retrace le raisonnement ayant mené à la solution. Elle substitue les éléments de contexte du raisonnement retrouvé par les éléments différents du contexte du nouveau cas, et rejoue le raisonnement dans ce nouveau contexte.
- *l'adaptation transformationnelle*, part du fait que nous avons des éléments de réponses à un problème qui nécessite des modifications selon des écarts de contexte observés entre le cas source et le nouveau cas. De ce fait, des éléments

de la solution du cas retrouvé sont modifiés, supprimés ou ajoutés grâce à un ensemble de règles d'adaptations.

- *l'adaptation compositionnelle*, utilise l'ensemble des cas similaires pour l'adaptation.

Beaucoup de systèmes se contentent d'une recopie simple de la solution du cas remémoré (Lenz et al., 1996), ou d'une composition des solutions de plusieurs cas remémorés (Netten et al., 1996). D'autres méthodes d'adaptation existent telles que la remémoration guidée par l'adaptation (Smyth et al., 1996), l'adaptation mémoire (Leake et al., 1996) ou encore l'analogie par dérivation et l'analogie par transformation (Veloso, 1995).

#### 2.2.4 Validation et mémorisation d'un cas

La phase de révision ou de validation de la solution proposée évalue la solution dans le monde réel (bilan d'un cas) et vérifie par l'introspection dans la base de cas la qualité des cas. Travaux développés dans la communauté RàPC sous le vocable de maintenance de la base de cas. Nous vous décrivons au paragraphe 3, l'algorithme que nous avons mis au point dans (Haouchine et al., 2007).

La mémorisation ou l'apprentissage d'un nouveau cas (**Cible**, **Sol(cible)**) ajoute le nouveau cas appris dans la base de cas, si celui-ci n'est pas déjà recensé dans la base de cas. Nous proposons dans ce papier un algorithme qui évalue cette incrémentation ou non du cas dans la base de cas.

### 3. Maintenance de la Base de Cas

La maintenance d'une base de cas est la mise en oeuvre de politiques permettant de réviser l'organisation et/ou le contenu de la base de cas afin d'améliorer le raisonnement futur (Leake et al., 1998). Cette maintenance est un ensemble de réalités différentes, qui va de la suppression des cas incohérents, à la sélection de groupes de cas permettant d'éliminer la redondance et d'améliorer le pouvoir de raisonnement du système, les cas peuvent être réécrit afin de réparer les problèmes d'incohérences (Smyth, 1998).

Il y a principalement deux stratégies de maintenance d'une base de cas :

- la première traite de problème d'optimisation, elle consiste à réduire la taille de la BC afin de diminuer le temps de recherche.
- La deuxième stratégie traite de partitionnement en plusieurs espaces de recherche ce qui permet de réduire ainsi le temps de recherche et de sélectionner de manière incrémentale les attributs qui sont riches en information et qui peuvent couvrir la structure de la BC entière.
- Une base de cas est de bonne qualité si elle permet au système de RàPC de résoudre le plus de problèmes possibles de manière correcte en un temps raisonnable (Zehraoui, 2004).



On note que la qualité d'un système de RàPC est liée à la définition et à la représentation d'un cas, l'organisation de la base de cas, les diverses indexations utilisées et la définition de "bonnes" mesures de similarités pour la recherche de cas, le lien recherche adaptation de cas.

### 3.1 Critères de bases

Notre système de ràPC, comporte une quarantaine de cas. Nous nous sommes intéressés aux stratégies d'élimination de cas de qualité moindre permet de réduire la base de cas tout en maintenant les critères de qualité de la base de cas.. Ce qui nécessite de définir des critères de qualité :

Nous décrivons ci-dessous deux critères permettant d'évaluer la qualité de la BC.

- La Compétence, qui est mesurée par le nombre de problèmes différents pour lesquels le système apporte une bonne solution (Smyth et al., 1995).
- La Performance, d'un système est mesurée par le temps de réponse qui lui est nécessaire pour proposer une solution à un cas cible (Smyth et al., 1995). Cette mesure est liée directement aux coûts d'adaptation et aux coûts de recherche.

Ces deux critères vont se traduire par deux notions de base défini par Smyth dans (Smyth et al., 1995). le recouvrement et l'atteignabilité d'un cas.

- Recouvrement d'un cas de la BC représente l'ensemble des cas cibles que ce cas peut résoudre

Nous noterons  $Vr(c)$  = cardinal de cet ensemble de cas cible associé au cas c

- Atteignabilité, d'un cas cible est l'ensemble de cas qui peuvent être utilisés pour le résoudre .

Nous noterons  $Va(c)$  = cardinal de cet ensemble de cas associé au cas cible c.

Smyth applique ces notions sur la partie problème du cas source (*srce*, *Sol(srce)*)

et bien entendu du cas cible (*cible*)

Ces deux notions sont utilisées pour évaluer la qualité d'une base de cas. Une base de cas a une bonne compétence lorsque son taux de recouvrement est élevé et le taux d'atteignabilité est faible. Une bonne compétence signifie que le nombre de problèmes pour lesquels la base de cas fournit une bonne solution et donc un bon diagnostic et l'action de réparation associée adéquate est élevé

- Nous avons proposé ainsi une Mesure de compétence prenant appui sur ces 2 notions ;

$$MesureCompétence(c) = \frac{Vr(c)}{Va(c)} \quad (1)$$

Ces notions vont nous permettre de catégoriser les cas en 4 types de cas qui seront à la base de l'élimination des cas non pivot.

### 3.2 Catégorisation de la base de cas suivant Smyth

Notre méthode prend appui sur la catégorisation proposée dans (Smyth et al., 1995), et notre mesure est inspirée de la mesure de recouvrement relatif RC dans (Smyth et al., 1998). Cette optimisation consiste à réduire la taille de la base de cas tout en ayant une compétence maximale.

Pour réorganiser la base de cas, nous avons exploité la catégorisation des cas par Smyth en 4 groupes liés à la performance de la base de cas. Et suivant le groupe d'appartenance le cas sera gardé, ou sera supprimé au détriment d'un représentant.

Soit une base de cas  $C = \{c_1, \dots, c_n\}$  et  $c \odot$  l'ensemble des cas cibles dans la base de cas.

Ainsi, 4 catégories de cas ont été défini par Smyth :

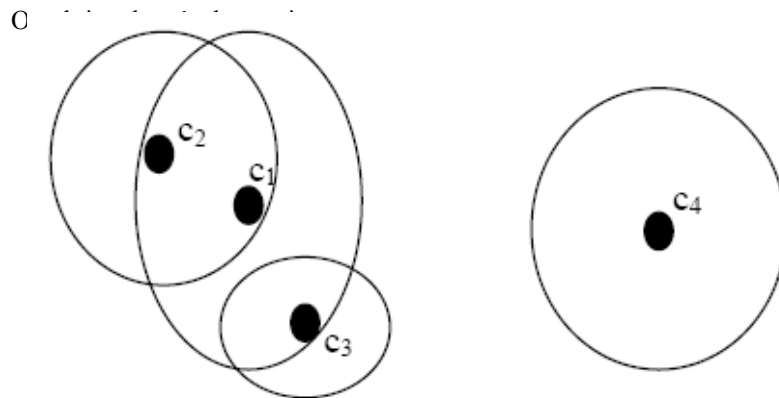
- **Pivot** : Un cas est dit pivot si son ensemble d'atteignabilité est réduit à un singleton (lui-même). Sa suppression réduit directement la compétence du système.  $\text{Pivot}(c) \text{ ssi } \text{Atteignabilité}(c) - \{c\} = \emptyset$ .

- **Support** : Un cas est dit de support s'il n'existe pas dans son ensemble d'atteignabilité des cas qui ont le même recouvrement.  $\text{Support}(c) \text{ ssi } c \odot \text{Atteignabilité}(c) - \{c\} : \text{Recouvrement}(c \odot) \text{ Couverture}(c)$

- **Auxiliaire** : Un cas est dit auxiliaire si son **recouvrement** est subsumé par le recouvrement d'un autre cas. Il n'affecte pas du tout la compétence du système.  $\text{Auxiliaire}(c) \text{ ssi } c \odot \text{Atteignabilité}(c) - \{c\} : \text{Recouvrement}(c) \text{ Recouvrement}(c \odot)$

- **Couverture** : Un cas est dit de couverture si son espace de recouvrement rencontre des régions dans les espaces de recouvrement des cas au sein de son ensemble d'atteignabilité. Ce cas n'affecte pas directement la compétence du système.  $\text{Couverture}(c) \text{ ssi } \text{Pivot}(c) \wedge \text{Recouvrement}(c) \cap \bigcup_{c \odot} \text{Atteignabilité}(c) - \{c\} \text{ Couverture}(c) \neq \emptyset$

Soit un exemple d'une base de cas contenant quatre cas, montrant l'espace de recouvrement et d'atteignabilité des cas  $c_1$ ,  $c_2$ ,  $c_3$  et  $c_4$  (figure 4).



**Figure 4.** Exemple sur le recouvrement et l'atteignabilité

$\text{Recouvrement}(c1)=\{c1, c2, c3\} \rightarrow \text{Vr}(c1)=3, \text{Atteignabilité}(c1)=\{c1, c2\} \rightarrow \text{Va}(c1)=2$   
 $\text{Recouvrement}(c2)=\{c1, c2\} \rightarrow \text{Vr}(c2)=2, \text{Atteignabilité}(c2)=\{c1, c2\} \rightarrow \text{Va}(c1)=2$   
 $\text{Recouvrement}(c2)=\{c1, c2\} \rightarrow \text{Vr}(c2)=2, \text{Atteignabilité}(c2)=\{c1, c2\} \rightarrow \text{Va}(c1)=2$   
 $\text{Recouvrement}(c2)=\{c1, c2\} \rightarrow \text{Vr}(c2)=2, \text{Atteignabilité}(c2)=\{c1, c2\} \rightarrow \text{Va}(c1)=2$

### 3.3 Algorithme de catégorisation et de suppression de cas

Nous avons exploité une partie de cette catégorisation, en ne prenant pas en compte les cas de couverture. Par contre, nous avons discriminés le cas auxiliaires en 2 types : les cas intra classe et les inter classe.

Les cas auxiliaires intra classe n'apportent rien par rapport aux cas pivots et aux cas de supports et peuvent être supprimés. Toutefois, on garde un cas auxiliaires inter classes à plus grande valeur de recouvrement. Les cas pivot ne doivent pas être touchés, on garde un représentant de chaque groupe de cas de support. Pour améliorer l'algorithme de catégorisation de Smyth, nous tenons compte de la valeur du MC dans le choix des cas de supports. Nous avons proposé dans (Haouchine et al., 2007) l'algorithme d'optimisation de la base de cas, tenant compte de cela, que nous avons comparé aux méthodes existantes du domaine et sur trois benchmarks différents.

---

#### Algorithme de la maintenance de la base de cas

---

1. Calcul des valeurs de recouvrement Vr et d'atteignabilité Va de chaque cas dans la BC.

2. Associer à chaque index du cas son ensemble de recouvrement et d'atteignabilité ( $\text{index\_cas} \leftarrow (\text{Vr}, \text{Va})$ ).

3. Déterminer les catégories des cas.

**Si cas de Support alors**

Classifier ces cas selon leurs valeurs de MC de manière incrémentée

**Pour** chaque groupe de support **faire**

Supprimer tout les cas sauf celui qui a la plus grande valeur MC

**FinPour**

**SinonSi** cas Auxiliaire intra-classe **alors**

Supprimer tout les cas

**SinonSi** cas Auxiliaires inter-classe **alors**

Supprimer tout les cas sauf celui qui a la plus grande valeur de MC

Sauvegarder l'index des cas supprimés

**SinonSi** cas Pivot **alors**

Ne pas supprimer

**FinSi**

4. Arrêt lorsque chaque cas recouvre seulement son propre cas parmi les cas existant dans sa classe.

---

#### Algorithme 1. Maintenance de la base de cas SORMEL

---

. Cette méthode donne de bons résultats et a permis d'assurer une taille minimale de la base de cas avec une compétence maximale par rapport aux trois bases de cas étudiées.

#### 4. Remémoration d'un cas

##### 4.1 Introduction

La remémoration d'un cas ou l'apprentissage de celui-ci se fait d'une manière incrémentale, car les cas résolus sont introduits automatiquement dans la base de cas s'ils remplissent les conditions nécessaires. Après avoir obtenu un cas adapté (cas pré-source), on obtient un cas cible dont sa partie problème et sa partie solution sont connues. Nous utiliserons les notions de recouvrement et d'atteignabilité. Mais nous calculerons ces notions non plus sur la partie problème du cas source et du cas cible, comme l'a défini Smyth, mais nous l'appliquerons également à la partie solution.

Le recouvrement d'un cas représente l'ensemble des cas similaire à ce cas.

Si on tient compte dans la mesure de similarité que des descripteurs de problème on obtient le Recouvrement partie problème ( $V_{rp}$ )

Si on tient compte dans la mesure de similarité que la partie solution on obtient le Recouvrement partie solution ( $V_{rs}$ )

Même chose pour l'atteignabilité,

Atteignabilité partie problème (Atteignabilité  $V_{ap}$ )

Atteignabilité partie solution ( $V_{as}$ )

Soit le recouvrement de la base de cas reflétant le taux de recouvrement moyen de toute la base de cas qui nous servira comme repère pour l'introduction d'un nouveau dans la base de cas.

$$V_{r BC} = \frac{\sum_{i=0}^n V_{r_i}}{n} \quad (2)$$

De ce fait, pour qu'un cas cible soit introduit dans la base de cas, il faut que la solution proposée par ce dernier n'existe pas dans la base de cas. Si elle existe, il faut que la partie problème (symptôme) soit atteignable par un nombre de cas inférieur au taux de moyen du recouvrement de la base de cas. Cette dernière condition assurera que le cas introduit contribuera à la compétence de la base de cas car son taux d'atteignabilité sera relativement faible.

L'algorithme d'apprentissage incrémental proposé est le suivant

## 4.2 Algorithme d'apprentissage incrémental

### *Algorithme d'apprentissage incrémental*

---

Soit la BC SORMEL // BC : Base de Cas

**Pour** chaque *cas pré-source* **faire**

**Si**  $V_{as} > 0$  **alors** // La solution du cas pré-source est atteignable par d'autres solutions des cas sources.

**Si**  $V_{ap} > V_{rBC}$  **alors**

            // La partie problème qui est atteignable par la partie problème des cas sources est supérieur à la moyenne du taux de recouvrement de la BC

$cas\ source \leftarrow cas\ cible$

$BC \leftarrow BC \cup cas\ source$

                // Introduire le cas à la BC

**FinSi**

**Sinon**

$cas\ source \leftarrow cas\ cible$  // Changement de statut du cas cible

$BC \leftarrow BC \cup cas\ source$  // Introduire le cas pré-source dans la BC

**FinSi**

**FinPour**

---

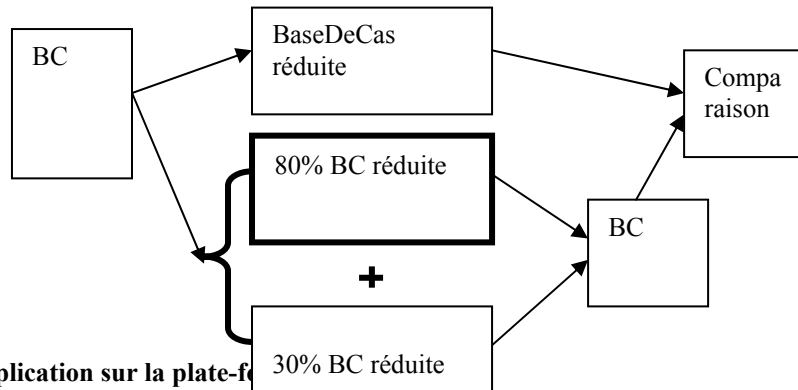
### **Algorithme 2.** *Apprentissage incrémental*

Conditionnellement aux taux d'atteignabilité de la partie solution du cas  $V_{as}$  le cas pré source n'existe pas dans la base de cas et doit y être intégré. Si  $V_{as}$  est égal à zero cela signifie qu'aucun cas similaire à ce nouveau cas pré-source n'a été recensé auparavant dans la base de cas. C'est un cas qui doit être ajouter dans la base de cas. On va donc changer de statut à ce cas pré-source en devenant cas source.

Par contre si la solution est atteignable  $V_{as} > 0$  alors on s'intéresse aux taux d'atteignabilité de la partie problème par rapport au taux d'atteignabilité moyen de la base de cas. pour être , ensuite introduit dans la base de cas. Ainsi, on obtient une base de cas qui est enrichie par ce nouveau cas appris.

## 4.3 Validation

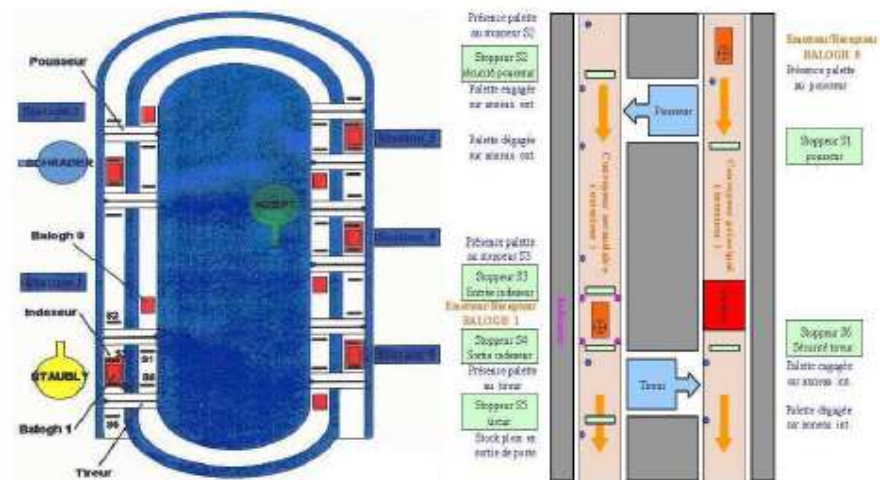
Soit une BaseDeCas. On applique l'algorithme de catégorisation des cas, on réduit la base de cas.



## 5. Application sur la plate-f

### 5.1. Description de la plate forme

Notre système d'aide au diagnostic et à la réparation a été mis en place sur une plate-forme industrielle d'un système de transfert de palettes SORMEL dont le modèle est présenté sur la figure 5. Le transfert SORMEL est un îlot flexible d'assemblage organisé en double anneau, constitué d'un système de transfert de palettes et de cinq stations de travail.



**Figure 5.** *Îlot flexible d'assemblage SORMEL (figure gauche) et composition d'un module du système de transfert (figure droite)*

Le déplacement des palettes est assuré par frottement sur des courroies, qui sont entraînées par des moteurs électriques. Les palettes sont munies d'une étiquette magnétique qui leur sert de « mémoire embarquée ». Ces mémoires peuvent être lues dans chaque station grâce à des plots magnétiques de lecture/écriture

(BALOGH) et permettent la mémorisation d'une gamme d'assemblage de produits. Ces étiquettes permettent donc de déterminer le cheminement des palettes à travers le système. Les palettes sont véhiculées sur l'anneau intérieur permettant ainsi le transit entre les différentes cellules. Lorsque l'une des palettes doit subir une opération de la part d'un robot (information lue sur l'étiquette de la palette), cette dernière est déviée sur l'anneau extérieur où se trouve le poste de travail concerné.

Chaque station de travail est étiquetée d'actionneurs pneumatiques (pousseurs, tireurs, indexeurs) et électriques (stoppeurs) ainsi que d'un certain nombre de capteurs inductifs (capteur de proximité D). La décomposition et l'analyse fonctionnelle de ce système permettent de créer une représentation d'un cas qui sera la brique de base pour le développement d'une base de cas.

Nous avons ainsi mis en place un programme en JAVA qui permet à partir d'une panne, de déclencher le cycle du RàPC. Après cela, il est capable de fournir le diagnostic d'une panne ainsi que l'action de réparation associée.

## 5.2. Elaboration d'un cas et de la base de cas

Pour modéliser les connaissances du domaine, des outils conventionnels tels que l'AMDEC, l'analyse fonctionnelle d'un équipement et les arbres de défaillances ont été utilisés. A partir de la décomposition de la plate-forme, avons développé un modèle qui sera à la base de la représentation du cas dans la base de cas.

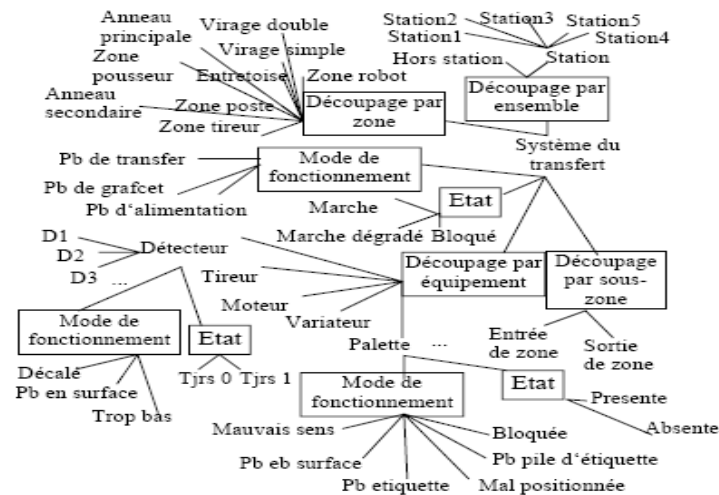


Figure 5. Modèle des descripteurs d'un cas (Rasovska et al., 2005)

| Cas (type 1)                                 | Attribut : valeur   |
|--|---|
| Description                                  |   |
| Symptôme                                     | Pb de transfert [pb d'alimentation, pb de grafcet, pb de transfert]   |
| Identification du contexte :<br>localisation | Localisation_ensemble : station [station, entretoise, virage-simple, virage_double]<br>Localisation_zone: pousseur [tireur, pousseur, indexeur(poste), conv-int, conv-ext]<br>Localisation_sous-zone : entrée [entrée, pousseur, sortie]  |
| Attribut du contexte : état                  | Détecteur D1 : 1 [0 (palette absente), 1 (palette présente)]<br>Balogh 0 : 1 [0 (palette n'entre pas dans la station), 1 (palette entre)]<br>Stoppeur S1 : 0 [0 (stoppeur en haut), 1 (stoppeur en bas)]<br>Pousseur : ne revient pas à sa position [pousse, ne revient pas à sa position, ne pousse pas ]<br>Symptôme pas de transfert indique : palette bloquée<br>Localisation de la palette bloquée : station, pousseur, entrée |
| Mode de fonctionnement<br>d'équipement       | D1 : bon fonctionnement<br>Balogh 0 : palette entre dans la station correspond au fonctionnement prévu<br>S1 : bon fonctionnement<br>Pousseur dégradé   |
| SOLUTION                                     | <b>Classe : POUSSEUR</b><br><b>DIAGNOSTIC: Vérin du pousseur hors service</b><br><b>ACTION : Remplacer le Vérin du pousseur</b>   |

**Tableau 1.** Exemple d'un cas

Le tableau 1 nous montre un exemple de cas

### 5.3. Remémoration et adaptation des cas

Dans cette phase, nous avons adopté l'algorithme des k plus proches voisins (kppv) en utilisant une mesure de similarité Euclidienne.

L'adaptation prendra appui sur un modèle du transfert qui associe à chaque classe de composant par exemple la classe de détecteur, suivant le positionnement de la palette le détecteur en cause. Cela permettra d'instancier un descripteur donné par un descripteur de la même famille et exploiter ces actions associées. On procède donc à une adaptation transformationnelle qui est destinée à manipuler une action (modifier, supprimer, ajouter une action).

Soit l'exemple suivant :

Une panne est détectée au niveau d'une des stations du transfert SORMEL. Les symptômes sont identifiés, ensuite cette panne va être mise sous forme de cas cible (voir partie problème sur la figure 6). Après avoir lancé l'algorithme de recherche, il nous fournit deux cas, le premier est similaire à 86% et le deuxième à 79%. Le premier cas sera sélectionné (voir figure 6) pour adapter son diagnostic par la suite.

Dans notre exemple, le cas remémoré à 86% appartient à la classe « détecteur ». Son diagnostic et son action de réparation sont en lien avec le descripteur « type du détecteur principal » qui a pour valeur « D4 ». Donc on va associer le diagnostic et l'action de réparation de ce cas source en instanciant la valeur du descripteur « type du détecteur principal » de la panne à diagnostiquer et à réparer.



Une fois que l'adaptation est faite, nous obtenons un cas cible dont son diagnostic et l'action de réparation associée sont connus (partie problème connue). On notera la solution obtenue *sol(cible)* permettant ainsi d'obtenir une fois validée (*Cible, Sol(cible)*)

#### 5.4. Résultats de la mise en œuvre des algorithmes de maintenance de la base de cas et de l'apprentissage incrémental

Nous précisons que dans notre étude, on utilise 3 catégories de cas à savoir les cas pivots, les cas de supports et les cas auxiliaires. Dans cette dernière catégorie, on a divisé ces cas en deux sous-catégories, les cas auxiliaires intra-classe, ceux qui concernent seulement les cas appartenant à la même classe des cas dans la base de cas. Ainsi que les cas auxiliaires inter-classe, ceux qui concernent des cas appartenant à des classes différentes.

##### Maintenance de la base de cas :

Au départ, on disposait d'une base de cas contenant 69 cas. Après avoir appliqué notre méthode, nous avons obtenu les résultats montrés au tableau 2 :

| Cas Pivots | Cas auxiliaires | Cas de supports | Groupes de supports |
|------------|-----------------|-----------------|---------------------|
| 16         | 4               | 49              | 19                  |

**Tableau 2.** Statistique de la base de cas SORMEL

Concernant les cas auxiliaires il a été recensé 2 cas inter-auxiliaires et 2 cas intra-auxiliaires. Après la suppression des 4 cas auxiliaires et des cas de supports en laissant seulement un cas de support dans chaque groupe de support (donc suppression de 30 cas de supports), on obtient les résultats suivants :

- Taille initiale de la base de cas = 69
- Taille de la base de cas obtenue = 35
- Taux de réduction = 49,27%
- Taux de compétence = 100%

De ce fait, on a obtenu une base de cas SORMEL réduite de pratiquement la moitié ayant la même compétence que la base de cas initiale. Ce qui implique qu'on obtient les mêmes résultats de diagnostic et d'actions de réparation associées avec un ensemble de symptômes réduit de presque la moitié. De ce fait, on obtient une base de cas optimale.

##### - Algorithme d'apprentissage

Nous avons à partir de la base de cas réduit de 35 cas extrait 2 ensembles

Un ensemble de 28 cas et un autre ensemble de 10 cas destiné à être ajouté ensuite, qui sera composé des cas restant plus d'autres cas du premier ensemble. En appliquant l'algorithme d'apprentissage incrémental proposé au paragraphe 3, nous avons obtenu les résultats suivants :

Sur 10 cas à apprendre, 7 cas ont été ajoutés à la base de cas et les 3 restants ont été jugés existants.

Pour comprendre la démarche, on s'appuie sur l'exemple précédent. On constate que le taux d'atteignabilité de la partie solution du cas diagnostiqué de l'exemple exposé précédemment est égal à 0 « Vas = 0 ». Ce qui signifie qu'aucun diagnostic similaire n'a été recensé auparavant dans la base de cas à celui de ce nouveau cas cible. De ce fait, ce cas cible va changer de statut et devient cas source, ensuite il est introduit dans la base de cas. Ainsi, on obtient une base de cas qui est enrichie par ce nouveau cas appris.

Donc, dans cette phase, le système fait une synthèse du raisonnement qui a été suivi de manière à sauvegarder les nouveaux cas pertinents qui permettront d'améliorer le raisonnement par la suite et d'enrichir la base de cas. Il en résulte une augmentation graduelle de la taille de la base de cas, ce qui met donc en évidence le besoin d'organiser et de maintenir la base de cas.

|   |  |
|---|--|
| C | Problème (symptôme)                              |
| A | - zone : anneau secondaire                       |
| S | - sous-zone : sortie                             |
|   | - composant-équipement : tapis extérieur         |
| C | - présence palette : oui                         |
| I | - type et détecteur principal : D7 = 0           |
| B | - stoppeur et son état : S5 = 0                  |
| L | - type et état du détecteur du contexte : D6 = 0 |
| E |  |
|   | Solution   |
|   | - Classe : Détecteur                             |
|   | - Diagnostic : problème surface détecteur D7     |
|   | - Action : nettoyé détecteur                     |

Figure 7. Panne mise sous forme de cas cible

|   |  |
|---|--|
| C | Problème (symptôme)                              |
| A | - zone : anneau secondaire                       |
| S | - sous-zone : convoyeur extérieur                |
|   | - composant-équipement : tapis extérieur         |
|   | - présence palette : oui                         |
|   | - type et détecteur principal : D4 = 0           |
|   | - stoppeur et son état : S2 = 1                  |
| 9 | - type et état du détecteur du contexte : D5 = 0 |
|   | Solution   |
|   | - Classe : Détecteur                             |
|   | - Diagnostic : problème surface détecteur D4     |
|   | - Action : nettoyé détecteur                     |

Figure 6. Cas source remémoré pour l'adaptation

## 5. Conclusion

Nous nous sommes intéressés dans cet article à un système d'aide au diagnostic et à la réparation et plus particulièrement aux dernières phases de ce système, à savoir la maintenance de la base de cas et l'apprentissage de cas dans le système. Notre algorithme de maintenance de la base de cas, qui dans la mouvance de Smyth, part de la catégorisation de la base, pour en déduire des cas non pertinents et nous a permis ainsi de réduire cette base de 45,27% sans altérer sa compétence.

En utilisant le même type de critères, un algorithme de remise à jour de cas dans la base de cas a été proposé tout en tenant compte de la structuration de la base de cas mis en place dans la phase de maintenance.

Notre étude sur le système de transfert de palettes SORMEL, nous conduit à étudier non plus le trajet d'une palette, mais le flux de palettes dans le transfert. Ce qui occasionnera dans le système de raisonnement à partir de cas certaines modifications.

## 6. Bibliographie

- Aamodt A., Plaza E., « Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches », *AI Communications*, vol.7, n°1, 1994, p. 39-59.
- Bagui S., et Pal N. R., « A multistage generalisation of the rank nearest neighbour classification rule », *Pattern Recognition Letters*, vol. 16, 1995, p. 601-614.
- Fuchs B., Représentation des connaissances pour le raisonnement à partir de cas : Le système ROCADE », Thèse de doctorat, Laboratoire Image Signal Acoustique de CPE Lyon, 1997.
- Giraud-Carrier C., « A Note on the Utility of Incremental Learning », *AI Communications*, vol.13, n°4, 2000, p. 215-223.
- Haouchine M.K., Chebel-Morello B., Zerhouni N., « Case Base Maintenance Approach », International Conference on Industrial Engineering Systems Management IESM'07, 2007.
- Leake D.B., Kinley A., Wilson D., « Linking Adaptation and Similarity Learning », *Proc. of the 8th Annual Conference of the Cognitive Science Society*, 1996.
- Leake, D.B., Wilson D.C., « Categorizing case-base maintenance: dimensions and directions. », *Advances in Case-Based Reasoning, 4th European Workshop, EWCBR 98, Proceedings*, Springer-Verlag, Berlin, Germany, 1998, p.196-207.
- Lenz M., Burkhard H. D., Pirk P., Auriol E., Manago M., « CBR for Diagnosis and Decision Support », *AI Communications*, vol. 9, n°3, 1996, p.138-146.
- Lenz M., Auriol E., Manago M., « Case-Based Reasoning Technology: From Foundations to Applications », *Diagnosis and Decision Support. Lenz M., Bartsch-Spörl, B., Burkhard, H.D. et Wess, S. (Eds.), Lecture Notes in Artificial Intelligence, Springer Verlag*, 1998.

- Lieber J. "Introduction p23-29 Raisonement à partir de cas collection Informatique et système d'information
- Main J., Dillon T.S., Shiu S.C.K., A tutorial on case based reasoning. *Soft Computing in Case Based Reasoning*, 2000, p. 1-28.
- Mille A., « Modèle conceptuel du raisonnement basé sur les cas », In *Isabelle Bichindaritz, édition, 4ème Séminaire Français sur le Raisonnement à Partir de Cas*, Paris, 1995, p. 40-52.
- Netten B. D., Vingerhoeds R. A., « Incremental Adaptation for Conceptual Design in edocs », In *Workshop on Adaptation in Case-Based Reasoning, ECAI-96*, Budapest, Hungary, 1996.
- Perron L., « La réutilisation de cas : une problématique commune à l'intelligence artificielle et à l'ergonomie cognitive, mais des points de vue différents ». *Ingénierie des connaissances*, 2000, p. 501-513.
- Rasovska I., Morello-Chebel B., Zerhouni « N. Développement d'une base de cas pour un système de raisonnement à partir de cas en maintenance industrielle ». PENTOM 2005, 2ème édition du colloque international francophone : Performance et Nouvelles Technologies en Maintenance, Marrakech, Maroc, 2005.
- Rodriguez J., Piechowiak S., « Expressing the behaviour of time-varying devices for the model based diagnosis ». *Proceedings of the 14th International Workshop on Expert Systems and their Applications*, Paris, 1994.
- Smyth B., Keane M.T., « Remembering To Forget: A competence Preserving Deletion Policy for Case-Based Reasoning Systems », In: *Proceeding of the 14th International Joint Conference on Artificial Intelligent*, Morgan-Kaufmann, 1995, P.377-382.
- Smyth B., Keane M. T., « Using Adaptation Knowledge to Retrieve and Adapt Cases », *Knowledge-Based Systems*, vol. 2, n° 9, 1996, p. 127-135.
- Smyth B., McKenna E., « Modelling the Competence of Case-Bases. Advances in case-based reasoning », *Lecture notes in computer science*, vol.488, 1998, p.23-25.
- Veloso M., et Aamodt A., « The 1st International Conference on Case-Based Reasoning », *ICCBR '95 – Sesimbra*, Portugal, Springer-Verlag, Berlin, Germany, 1995.